IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 1, 2021, accepted October 31, 2021, date of publication November 9, 2021,
date of current version November 18, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3126854

KNN-SC: Novel Spectral Clustering Algorithm Using k-Nearest Neighbors

JEONG-HUN KIM[1], JONG-HYEOK CHOI[2], YOUNG-HO PARK[3], CARSON KAI-SANG LEUNG[4], (Senior Member, IEEE), AND AZIZ NASRIDINOV[1]

[1]Department of Computer Science, Chungbuk National University, Cheongju 28644, South Korea
[2]Bigdata Research Institute, Chungbuk National University, Cheongju 28644, South Korea
[3]Department of IT Engineering, Sookmyung Women's University, Seoul 04310, South Korea
[4]Department of Computer Science, University of Manitoba, Winnipeg, MB R3E 0W2, Canada

Corresponding author: Aziz Nasridinov (aziz@chungbuk.ac.kr)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) by the Ministry of Education under Grant NRF-2021R1I1A3042145, and in part by the Institute for Information Communications Technology Promotion (IITP) Grant by the Korean Goverment through the Ministry of Science, ICT and Future Planning (MSIP) under Grant 2016-0-00406 (SIAT CCTV Could Platform).

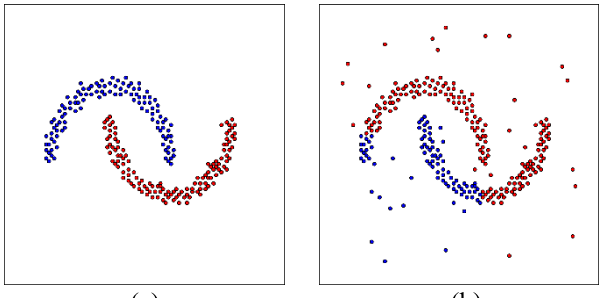
ABSTRACT Spectral clustering is a well-known graph-theoretic clustering algorithm. Although spectral clustering has several desirable advantages (such as the capability of discovering non-convex clusters and applicability to any data type), it often leads to incorrect clustering results because of high sensitivity to noise points. In this study, we propose a robust spectral clustering algorithm known as KNN-SC that can discover exact clusters by decreasing the influence of noise points. To achieve this goal, we present a novel approach that filters out potential noise points by estimating the density difference between data points using k-nearest neighbors. In addition, we introduce a novel method for generating a similarity graph in which various densities of data points are effectively represented by expanding the nearest neighbor graph. Experimental results on synthetic and real-world datasets demonstrate that KNN-SC achieves significant performance improvement over many state-of-the-art spectral clustering algorithms.

INDEX TERMS k-nearest neighbors, nearest neighbor graph, potential noise detection, spectral clustering.

I. INTRODUCTION

Clustering is an unsupervised data mining technique that partitions unlabeled data points into different groups based on their similarity. Over the last three decades, many clustering algorithms have been proposed, and these algorithms have achieved significant results in applications across multiple domains. We can categorize clustering algorithms into partitioning, hierarchical, graph-theoretic, model, and density-based approaches [1].

Considering the various clustering algorithms, spectral clustering [2] is a well-known graph-theoretic clustering algorithm. It generates a similarity graph for data points and embeds the data points into an eigenspace spanned by k eigenvectors through eigendecomposition on the similarity graph. By clustering the data points embedded in the eigenspace, non-convex clusters can be discovered. Particularly, spectral clustering can be easily applied to any data type because it relies only on the similarity graph [3]. Practically, it is widely applied in various fields such as network analysis [4], [5], computer vision [6], [7], and pattern recognition [8]–[11]. However, spectral clustering has a critical limitation that it leads to incorrect clusters because of high sensitivity to noise points [3].



(a) (b)

FIGURE 1. Performance of spectral clustering (SC) being affected by noise points: (a) discovery of exact clusters by SC; (b) failure of SC.

The associate editor coordinating the review of this manuscript and approving it for publication was Hocine Cherifi.

An example is illustrated in Fig. 1. For a dataset consisting of two moon shape clusters shown in Fig. 1a, the result of spectral clustering is exactly the same as the ground truth clusters. On the contrary, spectral clustering discovers completely incorrect clusters when some noise points are added to the same dataset as shown in Fig. 1b. Considering that noise points often define inter-cluster relationships that disturb exact clustering, these noise points corrupt a similarity graph of data points, and when the corrupted similarity graph embeds the data points into the eigenspace, spectral clustering leads to incorrect clustering results.

To address this issue, many researchers have proposed new clustering approaches, which are extensions of spectral clustering. One representative clustering approach [12]–[16] incorporates spectral clustering into density-based clustering. The density-based spectral clustering approach aims to discover clusters consisting of data points with similar densities. The clusters discovered by the density-based spectral clustering are robust to noise points because the noise points are typically sparse; therefore, they have no similar density to the data points included in the clusters. Consequently, the density-based spectral clustering approach discovers clusters that are robust to noise points using a similarity graph that represents the relationships between data points with similar densities. However, as the number of noise points increases, the density of the clusters is deformed, which may lead to incorrect clustering results. Other representative approaches [3], [17]–[19] proposed methods to learn a graph representation that minimizes the influence of noise points on the clustering result. To do this, many researchers have proposed optimization techniques for estimating and pruning noise points. However, similar to the density-based spectral clustering approaches, graph learning representation techniques are negatively influenced by the number of noise points. Specifically, the increased number of noise points reduces the sparsity of the data points and eventually corrupts the similarity graph, often leading to incorrect clustering results. Furthermore, because these approaches must assign all the data points to specific clusters, the noise points are treated as regular data points rather than outliers.

In this study, we propose a novel spectral clustering algorithm using k-nearest neighbors known as KNN-SC. KNN-SC first generates a nearest neighbor graph and uses a statistical method to estimate the density difference between vertices to filter the potential noise points. Thereafter, it expands the nearest neighbor graph based on the local density of each vertex to generate a density-based similarity graph, representing a density-based clustering structure. Finally, the KNN-SC discovers clusters using spectral clustering optimized to maximize the average density of clusters for the similarity graph. Therefore, it effectively improves the clustering performance and robustness against noise points by combining the advantages of the density-based spectral clustering and potential noise detection approaches. In addition, we demonstrate that the proposed method is robust against the number of noise points using extensive experiments.

The main contributions of this study are summarized below.

(1) By utilizing a density estimator based on the k-nearest neighbors and a statistical method, we can explicitly filter out potential noise points without the learning process of a graph representation.
(2) We introduce a novel method that generates a density-based similarity graph representing the adaptive density-based relationships between data points using the nearest neighbor graph and k-nearest neighbors.
(3) We propose a robust spectral clustering algorithm, KNN-SC, which overcomes the shortcomings of the existing algorithms.
(4) We provide experimental evaluations conducted on synthetic and real-world datasets to demonstrate the performance of the KNN-SC.

The rest of this study is organized as follows. Section II reviews the related studies. Section III introduces the foundational definitions and details of the proposed algorithm. Section IV compares the performance of KNN-SC to other clustering algorithms using several synthetic and real-world datasets, and Section V concludes the article.

II. RELATED STUDIES

In this section, we describe the basic concepts of spectral clustering and the definitions used in our study. Furthermore, we review the existing clustering algorithms to address the aforementioned issue of spectral clustering in Section I.

A. SPECTRAL CLUSTERING

Spectral clustering can be summarized in three steps [3]. First, it generates a similarity graph $G = (V, E)$. Whereas various strategies for generating the similarity graph exist, we focus on an ϵ-neighborhood graph using the radius ϵ. Thus, the set of edges E is defined as $\left\{(i,j)\,|d\left(v_i, v_j\right) \leq \epsilon, v_i, v_j \in V, 1 \leq i, j \leq m, i \neq j\right\}$, where $d\left(v_i, v_j\right)$ is the Euclidean distance between the vertices, V is the set of vertices, and m is the number of data points. Here, the similarity graph can be represented by a symmetric adjacency matrix $A \in \{0,1\}^{m \times m}$. Each element $a_{i,j}\,(1 \leq i, j \leq m)$ of the adjacency matrix A is 1 if $(i,j) \in E$; if otherwise, it is zero.

Second, a solution of the minimized ratio-cut or normalized cut for the similarity graph G, an objective of spectral clustering, is obtained. For example, an approximation of the ratio-cut is obtained using the trace minimization problem for the eigendecomposition of a Laplacian matrix L of the adjacency matrix A:

$$\begin{aligned} &\min_{H \in \mathbb{R}^{m \times k}} Tr\left(H^T L H\right) \\ &\text{s.t. } H^T H = I \\ &L = D - A \end{aligned} \tag{1}$$

where $Tr\,(\cdot)$ denotes the trace operator, m is the number of data points, k is the rank of the eigendecomposition, D is

a diagonal degree matrix whose elements are column-wise sums of A (i.e., $a_{ii} = \sum_j s_{ij}$), and $H \in \mathbb{R}^{m \times k}$ is the solution of the trace minimization problem in which the eigenvectors corresponding to the k-smallest eigenvalues of the Laplacian matrix L are concatenated. The normalization of A changes the Laplacian matrix L to a symmetric $\left(D^{-1/2}LD^{1/2}\right)$ or random walk $\left(D^{-1}L\right)$ Laplacian matrix.

Finally, the existing clustering algorithms (e.g., k-means [20]), are applied to the spectral embedding set H to discover the final clusters.

B. EXISTING ALGORITHMS

Various algorithms have been introduced to improve spectral clustering by alleviating issues related to noise points. One representative approach of these algorithms is to learn a graph representation (such as a similarity graph and an affinity matrix) to minimize the influence of noise points on the clustering result [3], [17]–[19], [21]–[23]. This approach strengthens the robustness against noise points by iteratively updating the graph representation based on the clustering results until an optimal solution is obtained. For example, Bojchevski *et al.* [3] proposed a robust spectral clustering algorithm (known as RSC) that minimized the influence of potential noise points by decomposing a similarity graph into two latent graphs: good and corrupted graphs. Specifically, they optimized a trace minimization problem on a good graph by updating the potential noise points corresponding to the corrupted graph. Thereafter, they minimized the influence of the potential noise points by performing spectral clustering on the good graph only. Other studies [17]-[19], [21] have proposed feature selection algorithms that minimized the influence of noise points using subspace learning. Zhu *et al.* [21] utilized the Frobenius norm and half-quadratic optimization to learn an affinity matrix from a low-dimensional space of the original data. This optimized affinity matrix represents an ideal clustering structure that removes the influence of noise points and redundant features.

The other approach is to incorporate ideas of the density-based clustering algorithms, such as utilizing a similarity computed by density estimation techniques [12]-[16], [24]. These algorithms [12], [14], [15], [24] reduce the influence of noise points by generating a similarity graph using the density-sensitive similarity. Beauchemin [13] proposed a new density-based similarity matrix through density estimation using k-means with subbagging. Hess *et al.* [16] presented a technique for optimizing the eigenvectors of a similarity graph to discover clusters with a large average density. By performing this optimization, an appropriate density for each cluster is automatically determined, and considering this process, the clustering structure becomes robust against noise points (see [16] for details).

Recently, researchers have proposed an approach that employs deep learning for spectral clustering [25]-[27]. For example, Tian *et al.* [25] proposed a deep learning-based spectral clustering algorithm that significantly improved the clustering performance by extracting latent representations using an autoencoder. Yang *et al.* [27] proposed a dual autoencoder network to extract robust latent representations of noise points and to discover the optimal clusters through deep spectral clustering.

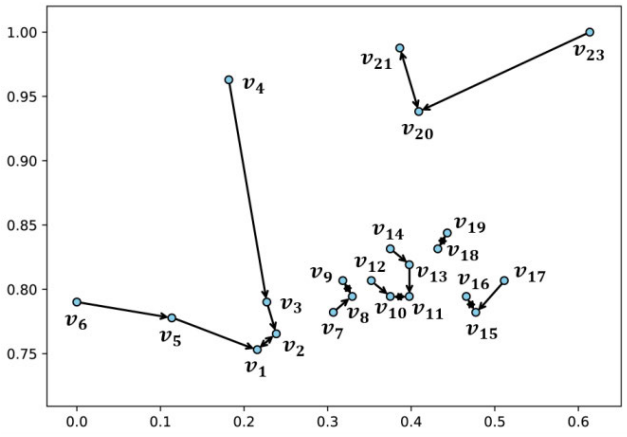


FIGURE 2. Example of the nearest neighbor graph, consisting of vertices corresponding to data points and directed edges from each vertex to its nearest neighbor.

III. PROPOSED ALGORITHM: KNN-SC

The proposed algorithm aims to generate an accurate similarity graph from the data points. As mentioned in Section II, traditional spectral clustering discovers clusters based on the eigenvectors obtained by decomposing the similarity graph. However, the similarity graph is easily corrupted by noise points, leading to incorrect clustering as shown in Fig. 1b. Therefore, considering the influence of noise points, we propose a novel approach that generates a density-based similarity graph using k-nearest neighbors to handle the noise points. We first identify locally dense data points as core vertices by utilizing the properties of the nearest neighbor graph. Thereafter, we filter out the potential noise points based on two assumptions. (i) The density of a noise point differs significantly from the average density of the data points, and (ii) the density of a noise point differs significantly from that of the neighbors that are not noise points. Subsequently, we generated a similarity graph by adaptively expanding the nearest neighbor graph based on the density of each core vertex. Because the similarity graph consists of vertices corresponding to the data points that are not filtered as potential noise points, the proposed algorithm can perform clustering by decreasing the influence of noise points.

A. SIMILARITY GRAPH GENERATION

First, we generate a nearest neighbor graph to identify locally dense data points as core vertices.

Let X be a set of m data points in the d-dimensional space of real values (i.e., $\forall x \in X : x \in \mathbb{R}^d$). The nearest neighbor

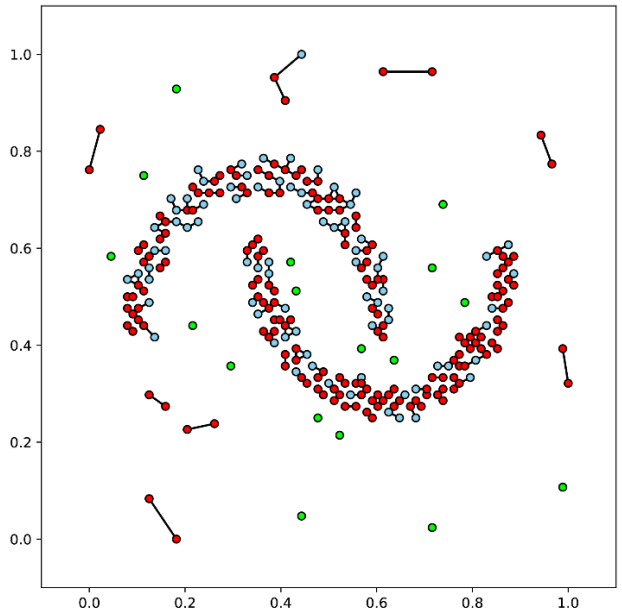


FIGURE 3. Example of the nearest neighbor graph in which the edges of locally sparse vertices are pruned.

graph is denoted by $G_{nn} = (V, E)$, where V is a set of vertices corresponding to the data points, E is a set of edges between each vertex $v_i \in V$ and its nearest neighbor, and the weight $w(i, j)$ of $(i, j) \in E$ is the Euclidean distance $d(v_i, v_j)$ between vertices v_i and v_j. Fig. 2 illustrates the nearest neighbor graph for some samples of the data points shown in Fig. 1b. As shown in Fig. 2, the nearest neighbor graph is composed of connected components that include at least one pair of vertices which are nearest neighbors to each other. For example, the nearest neighbor graph is composed of six connected components, and pairs of vertices (v_1, v_2), (v_8, v_9), (v_{10}, v_{11}), (v_{15}, v_{16}), (v_{18}, v_{19}), and (v_{20}, v_{21}) are nearest neighbors to each other. In addition, because such pairs of vertices have the smallest weights among the edges of each connected component, they can be locally dense data points. Based on the properties of the nearest neighbor graph, we define pairs of vertices that are nearest neighbors to each other as core vertices. On the contrary, considering the nearest neighbor graph, as the path from a vertex to a core vertex increases, this vertex becomes locally sparse because the distance from its nearest neighbor increases. Based on the density-based clustering paradigm that considers locally dense data points separated by locally sparse data points as clusters, we identify locally sparse vertices and prune all edges of these vertices from the nearest neighbor graph. To identify the locally sparse vertices, we utilize the *Z-test*, a statistical technique that probabilistically evaluates whether two sample sets are similar to each other in a normal distribution. Generally, because a dense vertex is close to its neighbors and a sparse vertex is opposite, it is possible to effectively identify a locally sparse vertex by comparing the distribution of k-nearest neighbors for each vertex. Let the set

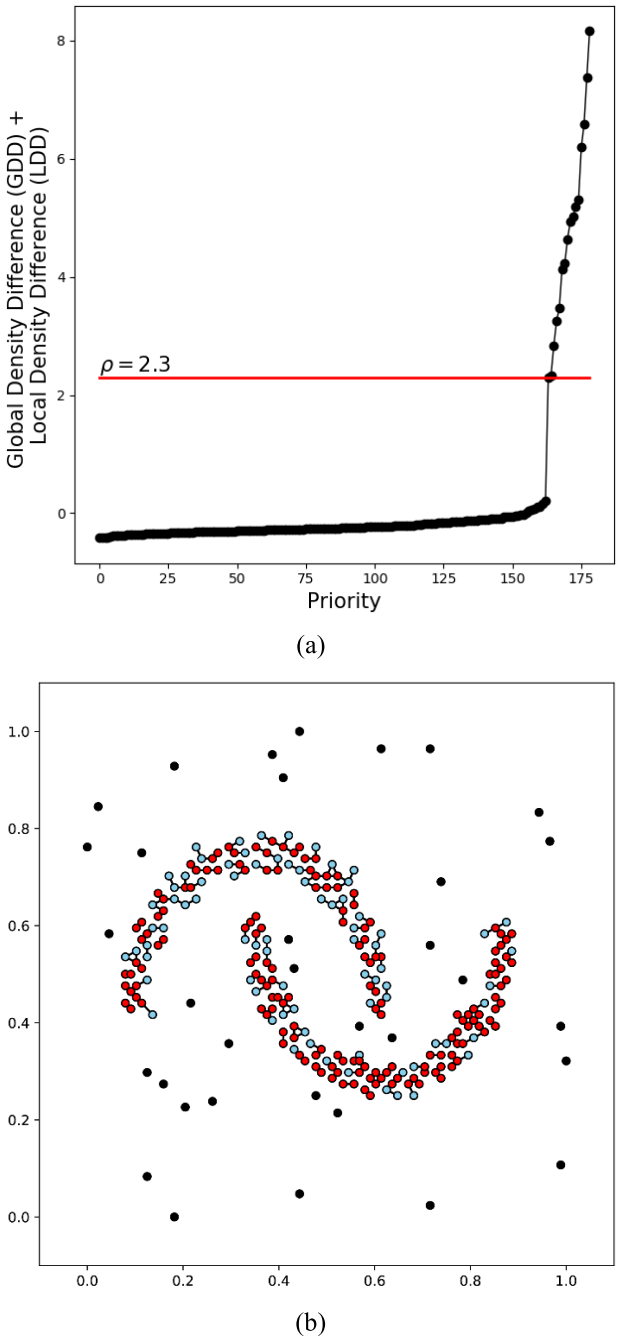


FIGURE 4. Example of filtering out high-priority potential noise based on a parameter ρ: (a) visualization for the result of aligning the core vertices according to the sum of *global density difference* and *local density difference*; (b) nearest neighbor graph with filtered connected components whose core vertices are potential noise points when the parameter $\rho = 2.3$.

of k-nearest neighbors of vertex v_i be N_i^k. The *Z-test* for two vertices (v_i and v_j) is defined as follows:

$$Z(v_i, v_j) = \frac{\sum_{u_i \in N_i^k} dist(v_i, u_i) - \sum_{u_j \in N_j^k} dist(v_j, u_j)}{k\sqrt{\sigma^2(v_i) + \sigma^2(v_j)}}, \quad (2)$$

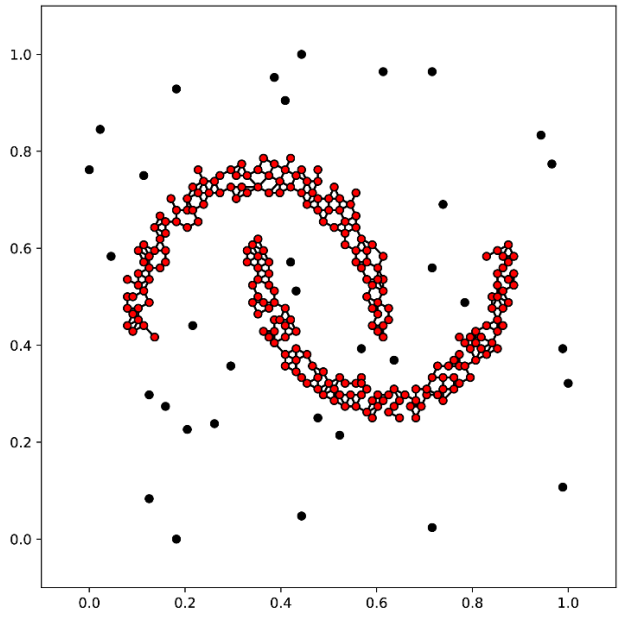


FIGURE 5. Visualization of the proposed similarity graph.

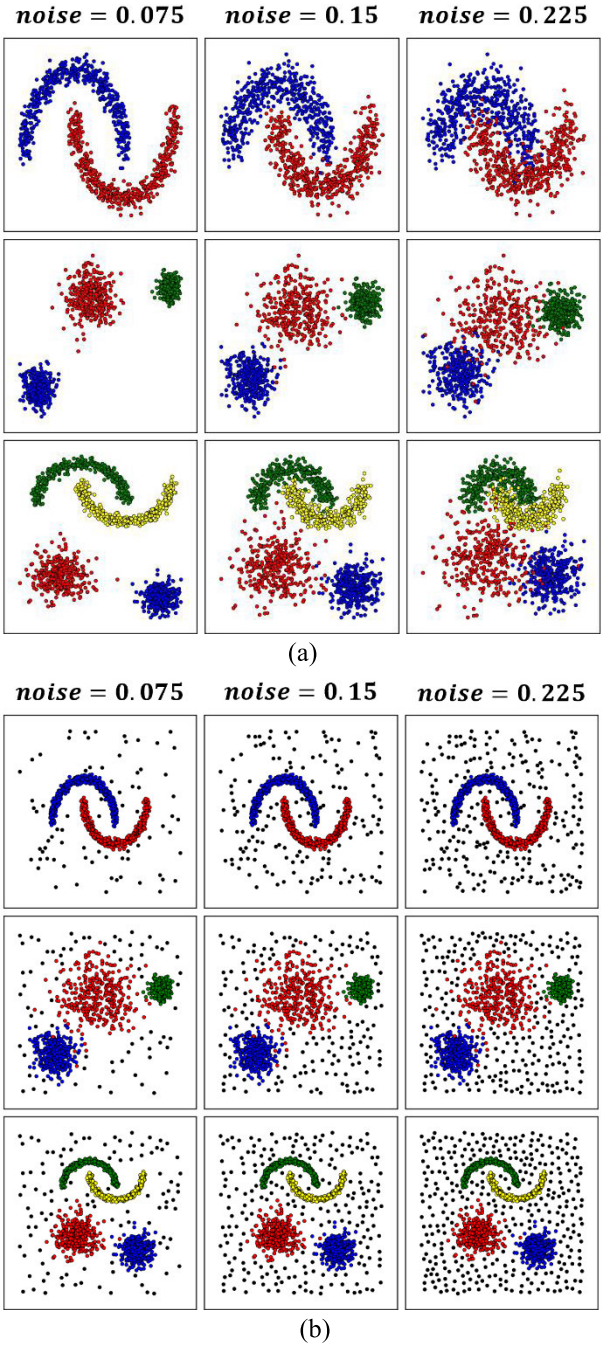


FIGURE 6. Visualization of moons, blobs, and mixed shape synthetic datasets based on the set values of a noise parameter: (a) examples of synthetic datasets with internal noise point; (b) examples of synthetic datasets with external noise points.

where $Z(\cdot)$ denotes a test statistic score, which is the result of the *Z-test* for the two vertices, and $\sigma^2(\cdot)$ is the standard deviation of the distances between the vertex and its k-nearest neighbors.

According to the statistical interpretation, when two sample sets are similar, Z is small, and in the opposite case, Z is large. Particularly, it is possible to calculate a *confidence interval (CI)* probabilistically, representing the statistical similarity of the two sample sets from Z. Typically, the two sample sets have a statistically significant similarity at a 95% confidence interval, indicating that Z is less than or equal to two. Hence, in this study, we consider all pairs of vertices for which Z is larger than two as dissimilar. Alternatively, we prune $\forall (i, j) \in E$, satisfying $\delta\left(v_i, v_j\right) > 2$. Fig. 3 shows the result of the pruning of the locally sparse vertices in the nearest neighbor graph generated for the data points in Fig. 1b. We have visualized the nearest-neighbor graph in an undirected graph to reduce the visual complexity. The red vertices represent the core vertices, the blue vertices indicate vertices with similar distance distributions for the core vertices, and the green vertices indicate locally sparse vertices identified by Eq. (2).

Generating the nearest neighbor graph for a dataset enables us to identify the connected components composed of data points with similar densities. These connected components correspond to the initial clusters of a given dataset. However, if there are noise points in the dataset, the connected components can be incorrectly generated for the noise points, considering them as core vertices. Because our proposed algorithm generates a density-based similarity graph by combining the connected components, noise points can corrupt the similarity graph. Therefore, we filter out the connected components whose potential noise points are the core vertices to address this corruption on the similarity graph. To filter out the potential noise points, we first define the local density of a vertex $v_i \in V$, denoted by $d(v_i)$, as the average distance from its k-nearest neighbors, indicating that $d(v_i) = \sum_{u_i \in N_i^k} dist(v_i, u_i) \Big/ k$. Thereafter, we filter out the potential noise points based on two assumptions. (i) The density of a

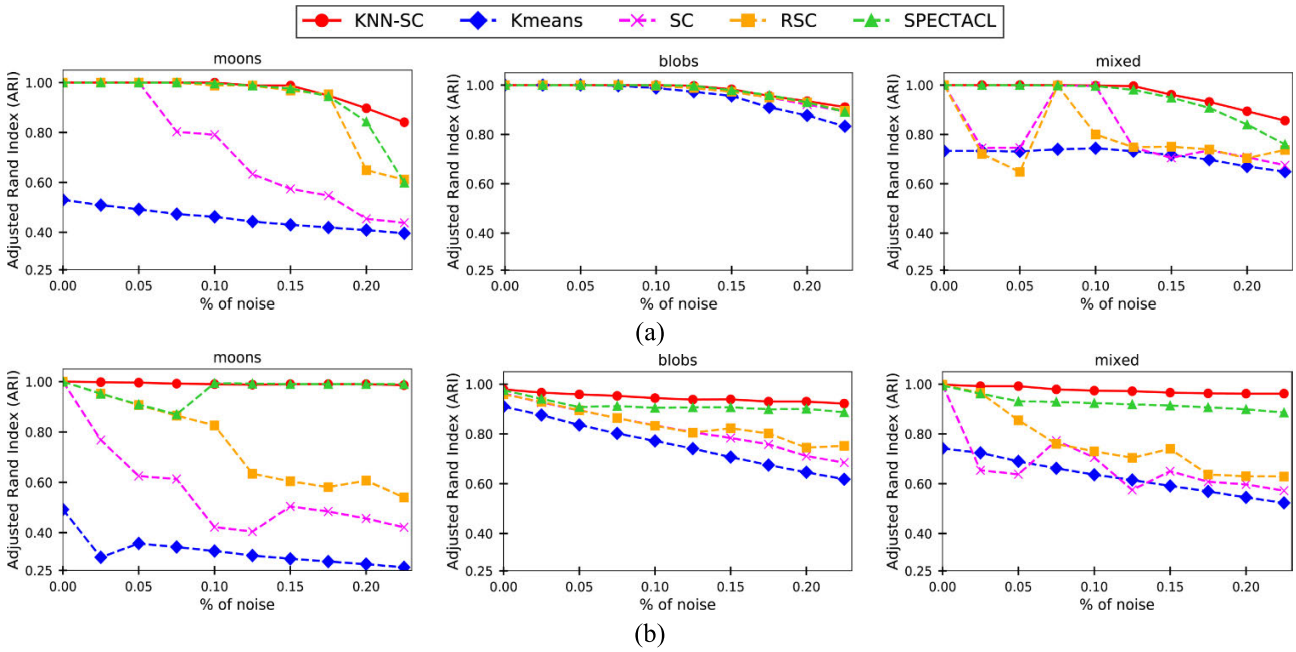


FIGURE 7. Comparison of the best ARIs (the higher, the better) of clustering algorithms based on the variation of a noise parameter: (a) case by internal noise point; (b) case by external noise points.

noise point differs significantly from the average density of data points because the noise point is relatively sparse; (ii) the density of a noise point differs significantly from that of neighbors that are not noise points. Therefore, we define two density difference measures: *global density difference (GDD)* and *local density difference (LDD)*. The *GDD* is the difference between the density of a vertex $v_i \in V$ and the average density of all the data points. *LDD* is the average density difference between a vertex $v_i \in V$ and its neighbors. The *GDD* and *LDD* are defined by the following equations:

$$GDD_i = \left(d\left(v_i\right) - d\left(\bar{X}\right)\right) \Big/ d\left(\bar{X}\right), \tag{3}$$

$$LDD_i = \frac{1}{k} \sum\nolimits_{u_i \in N_i^k} \frac{\left|d\left(v_i\right) - d\left(u_i\right)\right|}{d\left(u_i\right)}, \tag{4}$$

where $d\left(X\right) = \{d\left(x\right) | x \in X\}$, $d\left(\bar{X}\right)$ is the average of $d\left(X\right)$, and $|\cdot|$ is the absolute value function.

Because the vertices with large *GDDs* are sparse compared to those with small *GDDs* and the vertices with large *LDDs* are not similar to their neighbors, we can prioritize the potential noise points by aligning the core vertices using the sum of *GDD* and *LDD*. Considering the aligned core vertices, we sequentially filter out the high-priority potential noise points using the parameter ρ, which is the maximum threshold of the density difference between the data points that can create a cluster. The core vertices, which are not noise points, have small *GDDs* and *LDDs* because they are generally similar to each other, whereas the noise points have large *GDDs* and *LDDs*. Regarding this nature of the noise point, the parameter ρ can be easily determined the parameter ρ by selecting the sum of *GDD* and *LDD* with the largest gradient for the aligned core vertices.

Fig. 4 shows the result of filtering out the potential noise points by setting the parameter ρ to the sum of *GDD* and *LDD*, corresponding to the largest gradient. Considering Fig. 4a, all the core vertices of the nearest neighbor graph are prioritized by the sum of *GDD* and *LDD*, and the parameter ρ is set to 2.3, corresponding to the largest gradient indicated by the red line. Fig. 4b shows the nearest neighbor graph that filtered the potential noise points (black vertices) using the parameter ρ.

By filtering out the potential noise points, the connected components of the nearest neighbor graph are refined to consist of similar vertices. However, the nearest neighbor graph cannot sufficiently represent the similarity relationships between the vertices because there are no edges between the connected components. Therefore, we expand the nearest neighbor graph by adding new edges between the connected components with similar densities to generate a similarity graph. Let $G_{cc}^i = (V_i, E_i)$ be a connected component of the nearest neighbor graph $G_{nn} = (V, E)$ and v_{core}^i be its core vertex. We define the approximate density of a connected component as the density of its core vertex, that is, $d\left(v_{core}^i\right)$, because the vertices of the connected components are similar to each other. Thereafter, we add new edges to the two vertices (v^i and u), satisfying $dist\left(v^i, u\right) \leq d\left(v_{core}^i\right), v^i \in V_i, u \in V$, to expand the nearest neighbor graph. Each connected component expands adaptively according to its density, and the vertices connected by edges have similar densities. This expanded nearest neighbor graph is our proposed similarity graph, which can sufficiently represent the similarity relationships between vertices. Fig. 5 shows the similarity graph for the dataset shown in Fig. 4b. We used this graph to conduct spectral clustering.

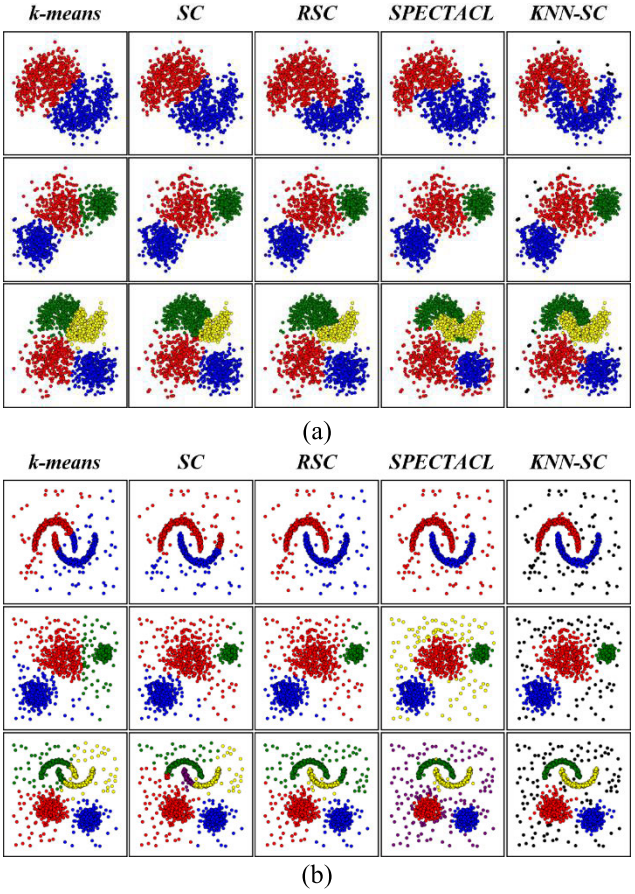


FIGURE 8. Example of filtering out high-priority potential noise based on a parameter ρ: (a) visualization for the result of aligning the core vertices according to the sum of global density difference and local density difference; (b) nearest neighbor graph with filtered connected components whose core vertices are potential noises when the parameter $\rho = 2.3$.

B. KNN-SC ALGORITHM

The clustering process of KNN-SC can be divided into three key steps: (i) filter out potential noise points from a dataset, (ii) generate a similarity graph, and (iii) apply optimized spectral clustering to discover clusters with the maximum average density.

Algorithm 1 describes the main procedures of KNN-SC and the detailed process of step (iii). We generate the proposed similarity graph in lines 1-2. Considering lines 3-5, we apply eigendecomposition after normalizing the similarity graph to calculate its eigenvalues λ and eigenvectors H. Regarding line 6, we discover optimal clusters from the similarity graph through *k-means* by utilizing a simple projection introduced in [16]. Algorithm 2 performs step (i) by finding the core vertices and connected components of the nearest neighbor graph and filtering out the potential noise points based on Eqs. (2), (3), and (4). Finally, Algorithm 3 fulfills step (ii) by adaptively expanding the nearest neighbor graph based on the local densities of the core vertices.

Algorithms 2 and 3 demonstrate the advantages of KNN-SC over the other spectral clustering algorithms because they use the properties of the nearest neighbor graph to alleviate the clustering corruption caused by the potential noise points and they reflect various densities of core vertices in the similarity graph.

Algorithm 1. KNN-SC

Input

X: the dataset;
k: the number of neighbors;
ρ: the maximum threshold;
c: the number of clusters;

Output

label: the list of cluster labels for each data point;

1: G_{nn} = NearestNeighborGraph(X, k, ρ)
2: A = SimilarityGraph(G_{nn})
3: D is the degree matrix of A
4: $L = D^{-1/2}(D - A)D^{1/2}$
5: λ, H = Eigendecomposition(L)
6: $label$ = *k-means*$\left(|\lambda|^{1/2}|H|, c\right)$
7: **return** *label*

IV. EXPERIMENTAL RESULTS

To illustrate the clustering results of KNN-SC, we conducted experiments on the synthetic and real-world datasets and compared the performance with those of other state-of-the-art clustering algorithms, including *k-means* [20], spectral clustering (SC) [2], RSC [3], and SPECTACL [16]. To ensure the validity of the experimental results, we used the *scikit-learn* Python library (*k-means* and SC) and the source code provided by the author (RSC and SPECTACL). KNN-SC was implemented in the Python programming language. All the algorithms were run on a machine that was equipped with a 3.2-GHz Intel CPU and 32 GB of memory, and the operating system was Windows 10 64 bit. To measure the clustering performance for each algorithm, we used two well-known evaluation metrics: normalized mutual information (NMI) [28] and adjusted rand index (ARI) [29]. The NMI is a measure used to evaluate clustering quality based on information theory. It is able to compare different clustering algorithms that have different numbers of clusters. However, the NMI may lead to erroneous evaluation because the number of clusters increases owing to the finite size effect [30]. Hence, we evaluated the clustering algorithms utilizing the ARI, which calculates the similarity between the ground truth labels and clustering results based on all the pairwise comparisons.

A. EXPERIMENTAL ANALYSIS OF SYNTHETIC DATASETS

Here, we evaluate the effectiveness of our algorithm using three synthetic datasets with different shapes. The three synthetic datasets were moons, blobs, and mixed shapes. These datasets were generated using the *scikit-learn* Python library. Particularly, to evaluate the robustness of our algorithm against noise points, we added *internal* and *external noise*

Algorithm 2. NearestNeighborGraph

Input

X: the dataset;
k: the number of neighbors;
ρ: the maximum threshold;

Output

G_{nn}: the nearest neighbor graph;

Initialize $G_{nn} = (V, E)$
for each $v_i \in V$ **do**
  $v_i.nn$ is the nearest neighbor of v_i
  $v_i.d = \sum_{u_i \in N_i^k} dist(v_i, u_i)/k$
  $\mathcal{D} \leftarrow v_i.d$
  $Z = \delta(v_i, v_i.nn)$ (Equation (2))
  if $Z \leq 2$ **then**
    if $v_i = v_i.nn.nn$ **then**
      $v_i.iscore = true$
    end if
  else
    $v_i.nn = \emptyset$
  end if
end for
for each $v_i \in V$ **do**
  if $v_i.iscore = true$ **then**
    $v_i.LDD = \frac{1}{k}\sum_{u_i \in N_i^k} \frac{|v_i.d - u_i.d|}{u_i.d}$ (Equation (4))
    $v_i.GDD = (v_i.d - \overline{\mathcal{D}})/\overline{\mathcal{D}}$ (Equation (3))
    if $v_i.LDD + v_i.GDD > \rho$ **then**
      $v_i.iscore = false$
      $v_i.nn = \emptyset$
    end if
  end if
end for
return G_{nn}

Algorithm 3. SimilarityGraph

Input

G_{nn}: the nearest neighbor graph;

Output

A: the adjacency matrix for the similarity graph;

for each $v_i \in V$ **do**
  if $v_i.iscore = true$ **then**
    $G_{cc}^i = (V_i, E_i)$ is the connected component for v_i
    $V_A \leftarrow V_i$
    $E_A \leftarrow E_i$
    for each $v_j^i \in V_i$ **then**
      for each $u \in V$ **do**
        if $dist(v_j^i, u) \leq v_i.d$ **then**
          $V_A \leftarrow u$
          $E_A \leftarrow (v_j^i, u)$
        end if
      end for
    end for
  end if
end for
A is the unweighted adjacency matrix for (V_A, E_A)
return A

points to each synthetic dataset. The *internal noise point* is Gaussian, as provided by the noise parameter of the *scikit-learn* Python library that adjusts the distribution of data points in a cluster. Considering each shape specification, we generated ten datasets by increasing the noise parameter from 0 to 0.225 in increments of 0.025. The *external noise points* are a set of random data points that are not included in any cluster. We also generated ten datasets for each shape specification by adding *external noise points* equal to the ratio of the noise parameter to the number of original data points. The moon and blob shape datasets have 1000 data points, and the mixed shape dataset has 1400 data points. Fig. 6 shows examples of the synthetic datasets with noise parameters of 0.075, 0.15, and 0.225. Fig. 6a shows the synthetic datasets with *internal noise points*. Because the noise parameter is larger, the data points of each cluster are widely distributed; therefore, the boundaries between the clusters are ambiguous. Fig. 6b shows the synthetic datasets with *external noise points*. As the noise parameter increases, the number of noise points (black points) also increases. These noise points corrupt the similarity graph, leading to incorrect clustering results.

To compare the best performance of the five clustering algorithms, we iteratively conducted experiments by increasing the parameters of each clustering algorithm. Considering SC, we adopted the traditional k-nearest neighbor graph as the similarity graph, and gradually increased the parameter k by one from 2 to 400. The RSC uses a parameter θ, a constraint for maximal number of corruptions. We also increased θ by one from 0 to 1000. SPECTACL uses an ϵ-neighborhood graph as the similarity graph, and we gradually increased the radius parameter ϵ by 0.001 from 0.001 to 3. Our KNN-SC uses two parameters, k and ρ. We increased k by one from 2 to 400, and set ρ to be the sum of *GDD* and *LDD* with the largest gradient of the aligned core vertices as mentioned in Section III-A. The number of clusters c for the five clustering algorithms, including *k-means*, was set to be equal to the ground truth ones. Through these iterations, we determined the parameters for which the clustering algorithms had the best ARI.

In Fig. 7, we demonstrate the performance of each clustering algorithm in terms of ARI against the two types of noise points. From Fig. 7, we can observe that KNN-SC typically achieves the highest ARI, showing the least variance of ARIs against the noise points. Although KNN-SC has a noticeable ARI degradation when the *internal noise points* are greater than 0.15, it is relatively robust to the noise points because the ARI fluctuates less than the other clustering algorithms. As Fig. 8a illustrates, KNN-SC detects the trajectory

FIGURE 9. Clustering result of KNN-SC on the Faces dataset. The clusters for the first hundred human faces are shown in different colors.

TABLE 1. Description of real-world datasets.

Dataset	Samples	Dimensions	Clusters
Iris	150	4	3
Banknote	1372	4	2
Wine	178	13	3
Ecoil	336	7	8
Seeds	210	7	3
Ionosphere	351	33	2
Sonar	208	60	2
Leaf	340	15	30
Vehicle	940	18	4
Faces	400	4096	40

of moons, whereas the other clustering algorithms cut both moons into half. We can also observe that all the algorithms achieve high ARIs on the blobs dataset, which is the easiest to cluster. Particularly interesting is the result of SPECTACL on a mixed dataset in which clusters of two shapes, such as moon and blob, are combined. We can easily observe that SPECTACL can detect the trajectories of the moons. However, because the similarity graph generated with a fixed radius ϵ is corrupted by *internal noise points*, blob-shaped clusters are also corrupted. On the contrary, KNN-SC is robust against *internal noise points* by generating a similarity graph with radius adaptive to the densities of the core vertices. In addition, because KNN-SC identifies and filters out *external noise points* as potential noise points, it has noise-independent performance as shown in Fig. 7b. In contrast, other clustering algorithms have lower ARIs as the number of noise points increases. The sole exception is the result of SPECTACL on the moons dataset with *external noise points*. As shown in Fig. 7b, SPECTACL has a lower ARI when the *external noise points* are less than 0.075 on the moons dataset. SPECTACL tends to regard *external noise points* as a cluster which can be seen from the clustering results for the blobs and mixed datasets in Fig. 8b; nevertheless; it often fails like the moons dataset.

B. EXPERIMENTAL ANALYSIS ON REAL-WORLD DATASETS

Here, we use ten real-world datasets of varying sizes, densities, and dimensionalities, and their characteristics are summarized in Table 1. These datasets were taken from the UCI dataset repository [31]. All the real-world datasets were normalized in advance.

TABLE 2. Comparison of competing algorithms on real-world datasets.

Dataset		KNN-SC	*k-means*	SC	RSC	SPECTACL
Iris	ARI	**0.875**	0.716	0.745	0.746	0.731
	NMI	**0.841**	0.742	0.778	0.798	0.722
Banknote	ARI	**0.995**	0.022	0.004	0.626	0.884
	NMI	**0.988**	0.017	0.016	0.611	0.835
Wine	ARI	**0.922**	0.899	0.869	0.882	0.831
	NMI	**0.910**	0.878	0.853	0.852	0.789
Ecoil	ARI	**0.750**	0.485	0.377	0.476	0.533
	NMI	**0.714**	0.618	0.576	0.601	0.586
Seeds	ARI	**0.810**	0.705	0.663	0.671	0.727
	NMI	**0.760**	0.674	0.673	0.658	0.673
Ionosphere	ARI	**0.645**	0.178	0.115	0.154	0.345
	NMI	**0.502**	0.135	0.076	0.117	0.325
Sonar	ARI	**0.161**	0.011	0	0	0.022
	NMI	**0.184**	0.012	0.001	0.002	0.086
Leaf	ARI	**0.371**	0.322	0.308	0.301	0.262
	NMI	**0.679**	0.672	0.656	0.651	0.596
Vehicle	ARI	**0.204**	0.076	0.106	0.107	0.131
	NMI	**0.219**	0.100	0.169	0.167	0.172
Faces	ARI	**0.570**	0.496	0.530	0.448	0.273
	NMI	**0.834**	0.801	0.815	0.774	0.669

To evaluate the clustering performance on real-world datasets, we used both evaluation metrics, ARI, and NMI. The parameters of the clustering algorithms were determined in a similar way to the experiments on synthetic datasets.

Table 2 summarizes the ARI and NMI for the five clustering algorithms on real-world datasets. According to Table 2, KNN-SC outperforms the other algorithms for all the real-world datasets. More importantly, KNN-SC has significantly higher ARI and NMI than the other clustering algorithms, regardless of the dimensionality of the dataset and the number of clusters.

Considering Fig. 9, KNN-SC can intuitively cluster similar human faces and detect noise points (denoted in gray color) using the direction of the face and gaze, or whether glasses are worn.

To evaluate the parameter sensitivity of KNN-SC, we also conducted experiments by changing the two parameters, k

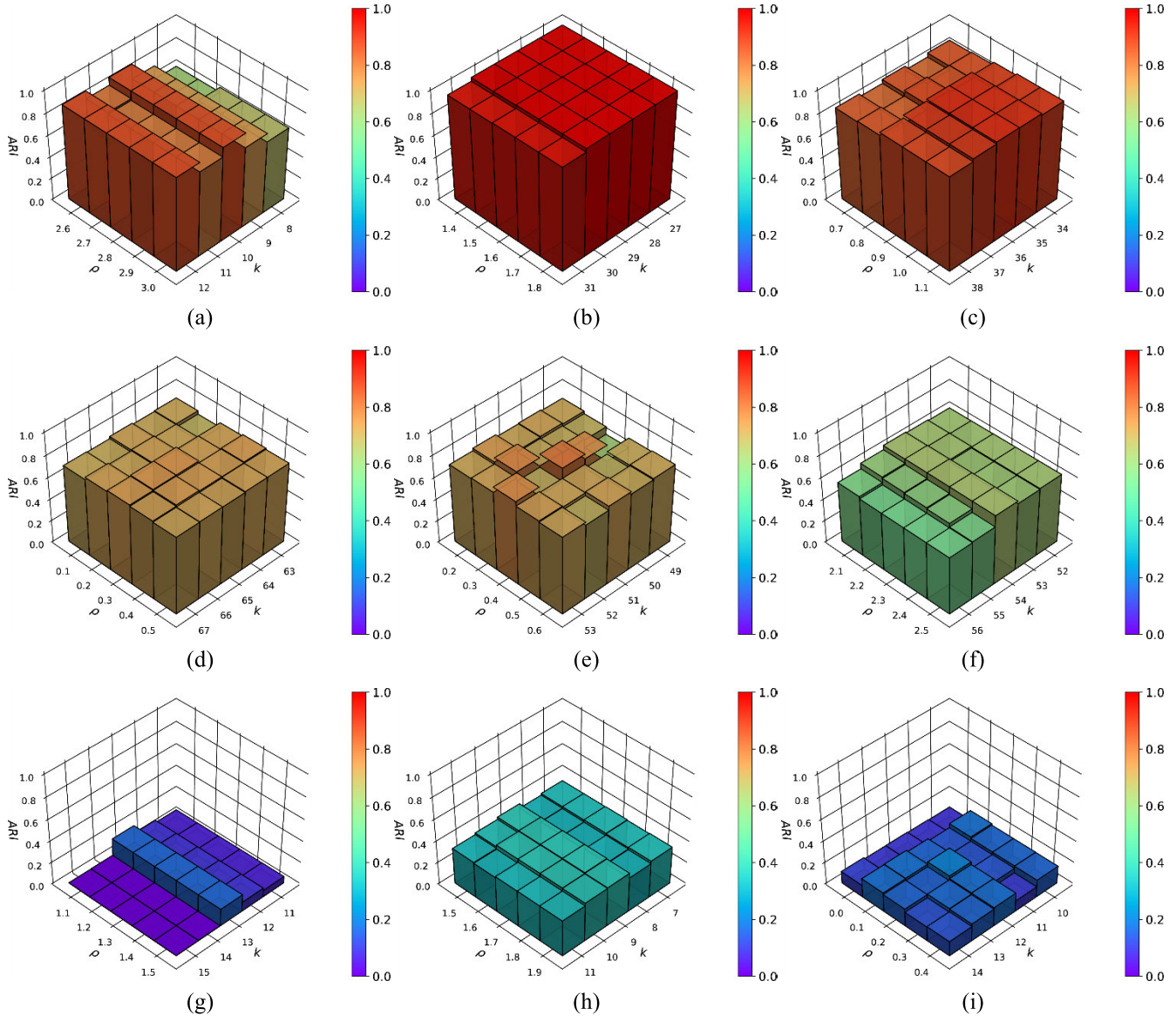


FIGURE 10. Parameter sensitivity of KNN-SC for nine real-world datasets: (a) Iris dataset; (b) Banknote dataset; (c) Wine dataset; (d) Ecoil dataset; (e) Seeds dataset; (f) Ionosphere dataset; (g) Sonar dataset; (h) Leaf dataset; and (i) Vehicle dataset.

and ρ, on nine real-world datasets, excluding the Faces dataset. Parameters k and ρ are increased by 1 and 0.1, respectively, until we achieve the best ARIs for real-world datasets.

Fig. 10 shows the experimental results for the parameter sensitivity of KNN-SC. There is no significant variance in the ARI, even if the parameters are changed for the seven datasets: Banknote, Wine, Ecoil, Seeds, Ionosphere, Leaf, and Vehicle. We can only observe that the ARI is notably changed on the Iris and Sonar datasets. Particularly, the ARI is more sensitive to parameter k than parameter ρ because parameter k determines the densities of the core vertices in KNN-SC, and a threshold parameter ρ is affected by the parameter k.

Considering these experiments, we observe that the KNN-SC is not sensitive to the parameters. In addition, if we determine parameter k first, then the parameter ρ can be easily determined.

V. CONCLUSION

In this article, we introduced a new spectral clustering algorithm known as KNN-SC, which is robust against noise points. Utilizing the properties of the nearest neighbor graph, we determine the locally dense data points that can represent the density variations of the dataset. Thereafter, we filter out the potential noise points that corrupt a similarity graph by estimating the difference in the density between the data points based on the k-nearest neighbors. Moreover,

we decrease the influence of noise points by generating a similarity graph representing the adaptive density-based relationships between data points. Therefore, we can significantly reduce the sensitivity of spectral clustering to noise points.

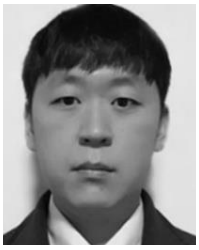
The results of comparative experiments show that, KNN-SC is the most robust to noise points on the synthetic datasets, competing with k-means, spectral clustering, RSC, and SPECTACL. Moreover, the superiority of KNN-SC over other clustering algorithms was demonstrated using several synthetic and real-world datasets. Particularly, the experiment on the Faces dataset illustrates the usefulness of KNN-SC in the field of computer vision.

In the future, we will optimize the proposed algorithm for extremely skewed or sparse datasets. Furthermore, we will apply KNN-SC to various application fields.

REFERENCES

[1] J. Lu and Q. Zhu, "An effective algorithm based on density clustering framework," *IEEE Access*, vol. 5, pp. 4991–5000, 2017, doi: 10.1109/ACCESS.2017.2688477.
[2] U. Von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, Aug. 2007, doi: 10.1007/s11222-007-9033-z.
[3] A. Bojchevski, Y. Matkovic, and S. Günnemann, "Robust spectral clustering for noisy data: Modeling sparse corruptions improves latent embeddings," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 737–746, doi: 10.1145/3097983.3098156.
[4] H. Van Lierde, T. W. S. Chow, and G. Chen, "Scalable spectral clustering for overlapping community detection in large-scale networks," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 4, pp. 754–767, Apr. 2020, doi: 10.1109/TKDE.2019.2892096.
[5] A. Karaaslanl and S. Aviyente, "Community detection in dynamic networks: Equivalence between stochastic blockmodels and evolutionary spectral clustering," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 7, pp. 130–143, 2021, doi: 10.1109/TSIPN.2021.3052047.
[6] T. Schultz and G. L. Kindlmann, "Open-box spectral clustering: Applications to medical image analysis," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 12, pp. 2100–2108, Dec. 2013, doi: 10.1109/TVCG.2013.181.
[7] M. D. Collins, J. Liu, J. Xu, L. Mukherjee, and V. Singh, "Spectral clustering with a convex regularizer on millions of images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, in Lecture Notes in Computer Science, Sep. 2014, pp. 282–298, doi: 10.1007/978-3-319-10578-9_19.
[8] H. Li, J. Liu, K. Wu, Z. Yang, R. W. Liu, and N. Xiong, "Spatio-temporal vessel trajectory clustering based on data mapping and density," *IEEE Access*, vol. 6, pp. 58939–58954, 2018, doi: 10.1109/ACCESS.2018.2866364.
[9] Y. Li and S. Wu, "Controlled islanding for a hybrid AC/DC grid with VSC-HVDC using semi-supervised spectral clustering," *IEEE Access*, vol. 7, pp. 10478–10490, 2019, doi: 10.1109/ACCESS.2018.2886533.
[10] S. Gupta and J. C. Rajapakse, "Iterative consensus spectral clustering improves detection of subject and group level brain functional modules," *Sci. Rep.*, vol. 10, no. 1, p. 7590, May 2020, doi: 10.1038/s41598-020-63552-0.
[11] C. Dinesh, S. Makonin, and I. V. Bajic, "Residential power forecasting based on affinity aggregation spectral clustering," *IEEE Access*, vol. 8, pp. 99431–99444, 2020, doi: 10.1109/ACCESS.2020.2997942.
[12] P. Yang, Q. Zhu, and B. Huang, "Spectral clustering with density sensitive similarity function," *Knowl.-Based Syst.*, vol. 24, no. 5, pp. 621–628, Jul. 2011, doi: 10.1016/j.knosys.2011.01.009.
[13] M. Beauchemin, "A density-based similarity matrix construction for spectral clustering," *Neurocomputing*, vol. 151, pp. 835–844, Mar. 2015, doi: 10.1016/j.neucom.2014.10.012.
[14] L. Wang, S. Ding, and H. Jia, "An improvement of spectral clustering via message passing and density sensitive similarity," *IEEE Access*, vol. 7, pp. 101054–101062, 2019, doi: 10.1109/ACCESS.2019.2929948.
[15] X. Tao, R. Wang, R. Chang, C. Li, R. Liu, and J. Zou, "Spectral clustering algorithm using density-sensitive distance measure with global and local consistencies," *Knowl.-Based Syst.*, vol. 170, pp. 26–42, Apr. 2019, doi: 10.1016/j.knosys.2019.01.026.
[16] S. Hess, W. Duivesteijn, P. Honysz, and K. Morik, "The SpectACl of nonconvex clustering: A spectral approach to density-based clustering," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Jul. 2019, pp. 3788–3795, doi: 10.1609/aaai.v33i01.33013788.
[17] X. Zhu, X. Li, S. Zhang, C. Ju, and X. Wu, "Robust joint graph sparse coding for unsupervised spectral feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 6, pp. 1263–1275, Jun. 2017, doi: 10.1109/TNNLS.2016.2521602.
[18] X. Zhu, S. Zhang, Y. Li, J. Zhang, L. Yang, and Y. Fang, "Low-rank sparse subspace for spectral clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 8, pp. 1532–1543, Aug. 2018, doi: 10.1109/TKDE.2018.2858782.
[19] X. Zhu, S. Zhang, W. He, R. Hu, C. Lei, and P. Zhu, "One-step multi-view spectral clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 10, pp. 2022–2034, Oct. 2018, doi: 10.1109/TKDE.2018.2873378.
[20] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient K-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002, doi: 10.1109/TPAMI.2002.1017616.
[21] X. Zhu, J. Gan, G. Lu, J. Li, and S. Zhang, "Spectral clustering via half-quadratic optimization," *World Wide Web*, vol. 23, pp. 1969–1988, Nov. 2020, doi: 10.1007/s11280-019-00731-8.
[22] Z. Kang, C. Peng, Q. Cheng, and Z. Xu, "Unified spectral clustering with optimal graph," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Apr. 2018, pp. 3366–3373.
[23] F. Nie, X. Wang, C. Deng, and H. Huang, "Learning a structured optimal bipartite graph for co-clustering," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2017, pp. 4132–4141.
[24] Y. Wang, S. Ding, L. Wang, and L. Ding, "An improved density-based adaptive p-spectral clustering algorithm," *Int. J. Mach. Learn. Cybern.*, vol. 12, pp. 1–12, Nov. 2020.
[25] F. Tian, B. Gao, Q. Cui, E. Chen, and T. Y. Liu, "Learning deep representations for graph clustering," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Jun. 2014, pp. 1293–1299.
[26] U. Shaham, K. Stanton, H. Li, B. Nadler, R. Basri, and Y. Kluger, "SpectralNet: Spectral clustering using deep neural networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2018, pp. 1–20.
[27] X. Yang, C. Deng, F. Zheng, J. Yan, and W. Liu, "Deep spectral clustering using dual autoencoder network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4066–4075.
[28] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Dec. 2002.
[29] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. Amer. Statist. Assoc.*, vol. 66, no. 336, pp. 846–850, Dec. 1971, doi: 10.1080/01621459.1971.10482356.
[30] P. Zhang, "Evaluating accuracy of community detection using the relative normalized mutual information," *J. Stat. Mech. Theor. Exp.*, vol. 2015, no. 11, p. 11006, Nov. 2015, doi: 10.1088/1742-5468/2015/11/P11006.
[31] D. Dua and C. Graff. (2019). UCI Machine Learning Repository. School of Information and Computer Science. University of California. Irvine, CA, USA. [Online]. Available: http://archive.ics.uci.edu/ml/

JEONG-HUN KIM received the B.Sc. and M.Sc. degrees in computer science from Chungbuk National University, South Korea, in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree in computer science. His research interests include traditional databases, machine learning, data mining, manufacturing optimization in smart factory, artificial intelligence, computer vision, and human action recognition in the wild.



JONG-HYEOK CHOI received the B.Sc. degree in computer education and the M.Sc. and Ph.D. degrees in computer science from Chungbuk National University, South Korea, in 2015, 2017, and 2021, respectively. He is currently a member of the Data Analysis Laboratory led by Prof. Aziz Nasridinov and a Postdoctoral Researcher at the Bigdata Research Institute, Chungbuk National University. His research interests include traditional databases, big data analysis, artificial intelligence, computer vision, and e-learning solution for K-12 students and teachers.

YOUNG-HO PARK received the B.Sc. and M.Sc. degrees in computer engineering from Dongguk University, in 1990 and 1992, respectively, and the Ph.D. degree from the Department of Computer Science, Korea Advanced Institute of Science and Technology (KAIST), in 2005. His Ph.D. research includes efficient query processing in heterogeneous XML documents. He was a Senior Research Staff at the ISDN Administration and Maintenance Division for TDX-10 ISDN, the Real-Time DBMS Division, and the Real-Time Operating System Division, Electronics and Telecommunication Research Institute (ETRI), from 1993 to 1999. He was also a Postdoctoral Researcher with the Advanced Information Technology Research Center (AITrc), KAIST, from 2005 to 2006, after receiving the Ph.D. degree. He is currently a Professor with the Department of IT Engineering, Engineering School, Sookmyung Women's University. Recently, his research interests include data science based on data analytics, data management systems (DBMS), information retrieval (IR), machine learning (ML), XML, and IT convergence with other fields, such as music, design, economy, business management, advertisement, and bio-informatics.

CARSON KAI-SANG LEUNG (Senior Member, IEEE) received the B.Sc. (Hons.), M.Sc., and Ph.D. degrees from The University of British Columbia, Canada. He is currently a Full Professor with the Department of Computer Science, University of Manitoba, Canada. He is the Founder and the Director of the Database and Data Mining Laboratory. His research interests include data mining and analysis, machine learning, data science, big data, databases (including image databases), data management, data warehousing, data visualization and visual analytics, bioinformatics, health informatics and electronic health, web technology and services, social computing, and social network analysis. He has contributed more than 300 refereed publications in these areas. He has served on organizing committees for international conferences, like ACM SIGMOD, IEEE ICDM, IEEE DSAA, IEEE/ACM ASONAM, and ACM CIKM. He has also served on the Steering Committee for IEEE SmartData and the Advisory Committee for IEEE CBDCom. Moreover, he has served as an Associate Editor for Springer journals, like *Social Network Analysis and Mining* (SNAM) and *Network Modeling Analysis in Health Informatics and Bioinformatics* (NetMAHIB).

AZIZ NASRIDINOV received the B.Sc. degree from the Tashkent University of Information Technologies, Uzbekistan, in 2006, and the M.Sc. and Ph.D. degrees in computer engineering from Dongguk University, in 2009 and 2012, respectively. He is currently an Associate Professor with the Department of Computer Science, Chungbuk National University, South Korea. His research interests include traditional databases, big data analytics with machine learning, and computer vision. He has served as a Program Committee Member and a Co-Organizer for numerous top-tier conferences, including IEEE Big Data, AAAI, and CHI. He also served on the editorial board for several international journals.